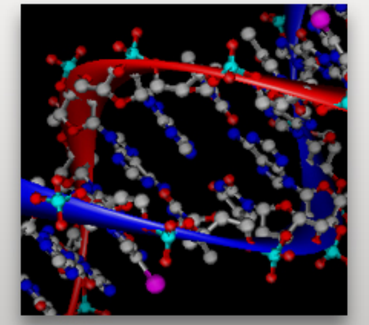
What’s New with File System

Session 506

These are confidential sessions—please refrain from filming or taking pictures.



OS Foundations

What’s New with File System

Chris Emura
Filesystem Engineering Manager

http://developer.apple.com/wwdc2005/

What You'll Learn

- *Symmetric MultiProcessor / Kernel Programming Interface*
- Extended Attributes
- Access Control Lists
- File Notification
- Copy Engine
- Oldies But Goodies

And plenty of time for Q&A…

Intended Audience

SMP/KPI Changes

http://developer.apple.com/wwdc2005/

SMP/KPI

- From 2 locks (funnels) to fine grained locking model
- Sustainable kernel programming interface
- kpi-feedback@apple.com
- Lab

Related lab

Writing File Systems Using Supported KPIs

Lab 5001
Wednesday 3:30pm

SMP/KPI

- Plan to drop funnel support in Chablis
- F_NOCACHE expanded to per-fd as well as per-vnode

Extended Attributes

http://developer.apple.com/wwdc2005/

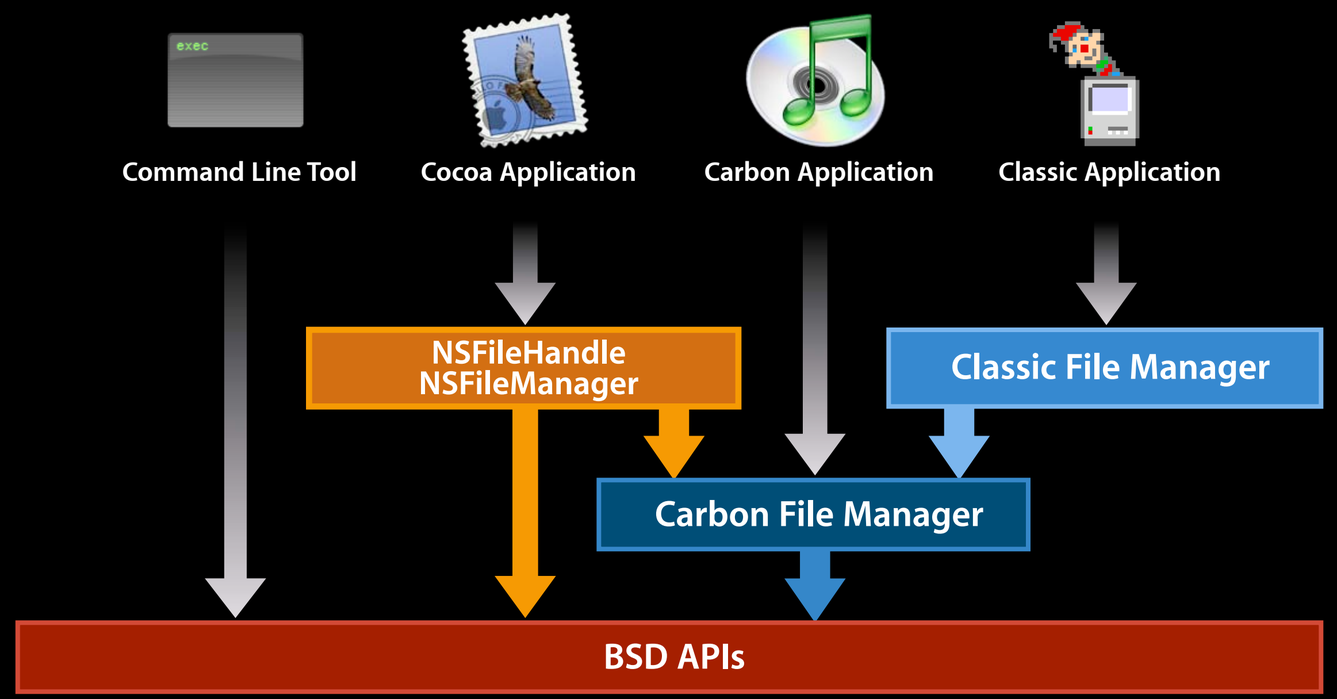
Extended Attributes

Overview

- Supported on all filesystems
- Reverse-DNS name convention (eg. com.apple.test.myApp.md5)
- Be aware of current limitations

Technology Frameworks

- Cocoa—Object oriented
- Carbon—Procedural
- BSD—Lower level support for all operations



Extended Attributes

listxattr(2)

```
/* get the size of our EA names and allocate some space */
nsize = listxattr(path, NULL, 0, 0);
if (NULL == (attrnamebuf = (char *)malloc(nsize))) {
        fprintf(stderr, "malloc() failed, path=%s nsize=%ld\n",
            path, nsize);
        return;
}

/* get the list of EA names */
lsize = listxattr(path, attrnamebuf, nsize, 0);

if (lsize >= 0) {
```

From ‘lsx.c’ sample code

Extended Attributes

getxattr(2)

```
/* get the size of the current EA */
asize = getxattr(path, ptr, NULL, 0, 0, 0);
if (asize < 0) {
    fprintf(stderr, "getxattr returned: %ld (%s)\n",
    asize, strerror(errno));
    break;
}
if (asize < sizeof(attrdatabuf)) {
    ssize_t asize2;
    asize2 = getxattr(path, ptr, attrdatabuf, sizeof(attrdatabuf), 0, 0);
```

From ‘lsx.c’ sample code

Extended Attributes

setxattr(2)

```
for (i=3 ; i<argc ; i++) {
    path = argv[i];
    if (0 != setxattr(path, EAname, EAvalue, strlen(EAvalue)+1,
    0, 0)) {
        fprintf(stdout, "setxattr() on %s: %s\n", path,
        strerror(errno));
    }
}
```

From ‘setx.c’ sample code

Extended Attributes

removexattr(2)

```
for (i=2 ; i<argc ; i++) {
    path = argv[i];
    if (0 != removexattr(path, EAname, 0)) {
        fprintf(stdout, "removexattr() on %s: %s\n", argv[i],
        strerror(errno));
    }
}
```

From ‘rmx.c’ sample code

Extended Attributes

Preservation

- Be mindful, even if you don’t use EAs directly
- Copy, move, archive, “safe save”
  - Data
  - EAs
  - Security information
- Industry fragmentation
- Tools, utilities, applications
  - Finder
  - cp, mv, scp, emacs, vim, pico, cpio, tar, rsync, gzip, bzip2

Extended Attributes

rename(2) semantics

Start with files ‘foo’ and ‘bar’…

foo

data
```
I am foo data.
```

EA
```
com.apple.emura.demo.fooEA
```

bar

data
```
I am bar data.
```

EA
```
com.apple.emura.demo.barEA
```

After a rename() of ‘foo’ to ‘bar’…

bar

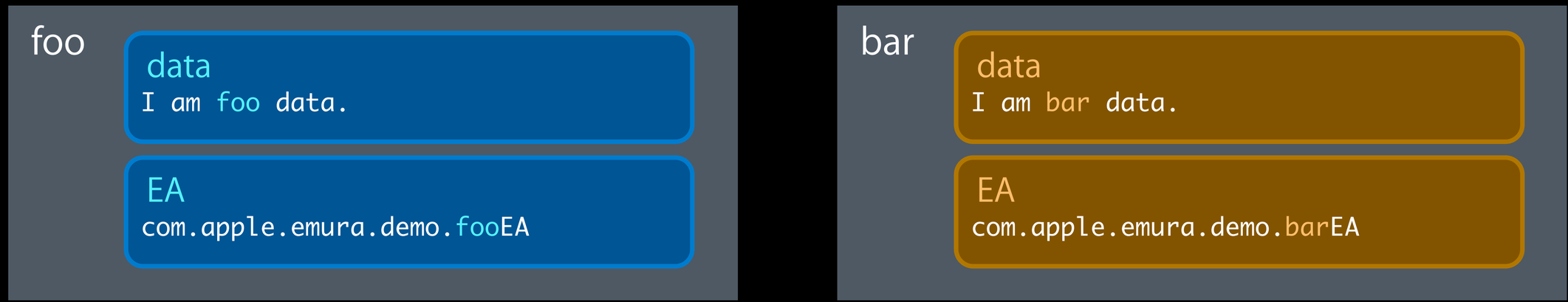
data
```
I am foo data.
```

EA
```
com.apple.emura.demo.fooEA
```

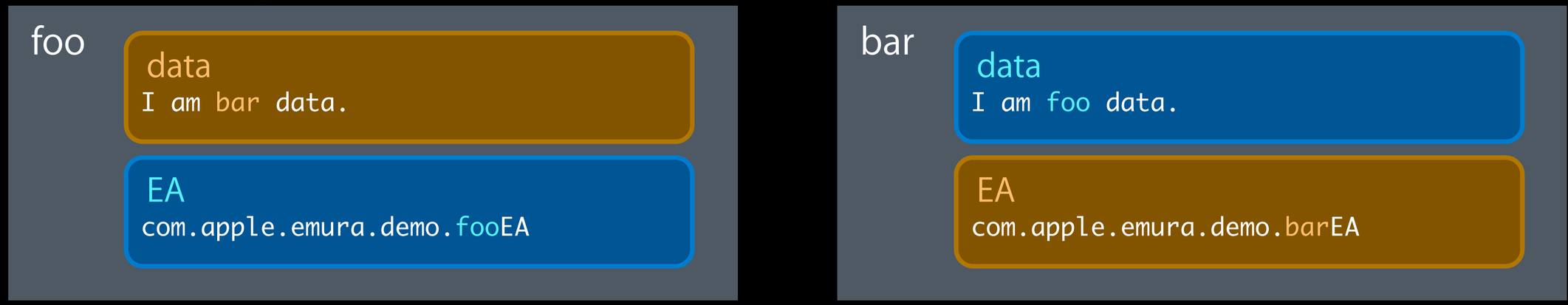
Extended Attributes

exchangedata(2) semantics

Start with files ‘foo’ and ‘bar’…



After a exchangedata() on ‘foo’ and ‘bar’…



Access Control Lists

http://developer.apple.com/wwdc2005/

Access Control Lists

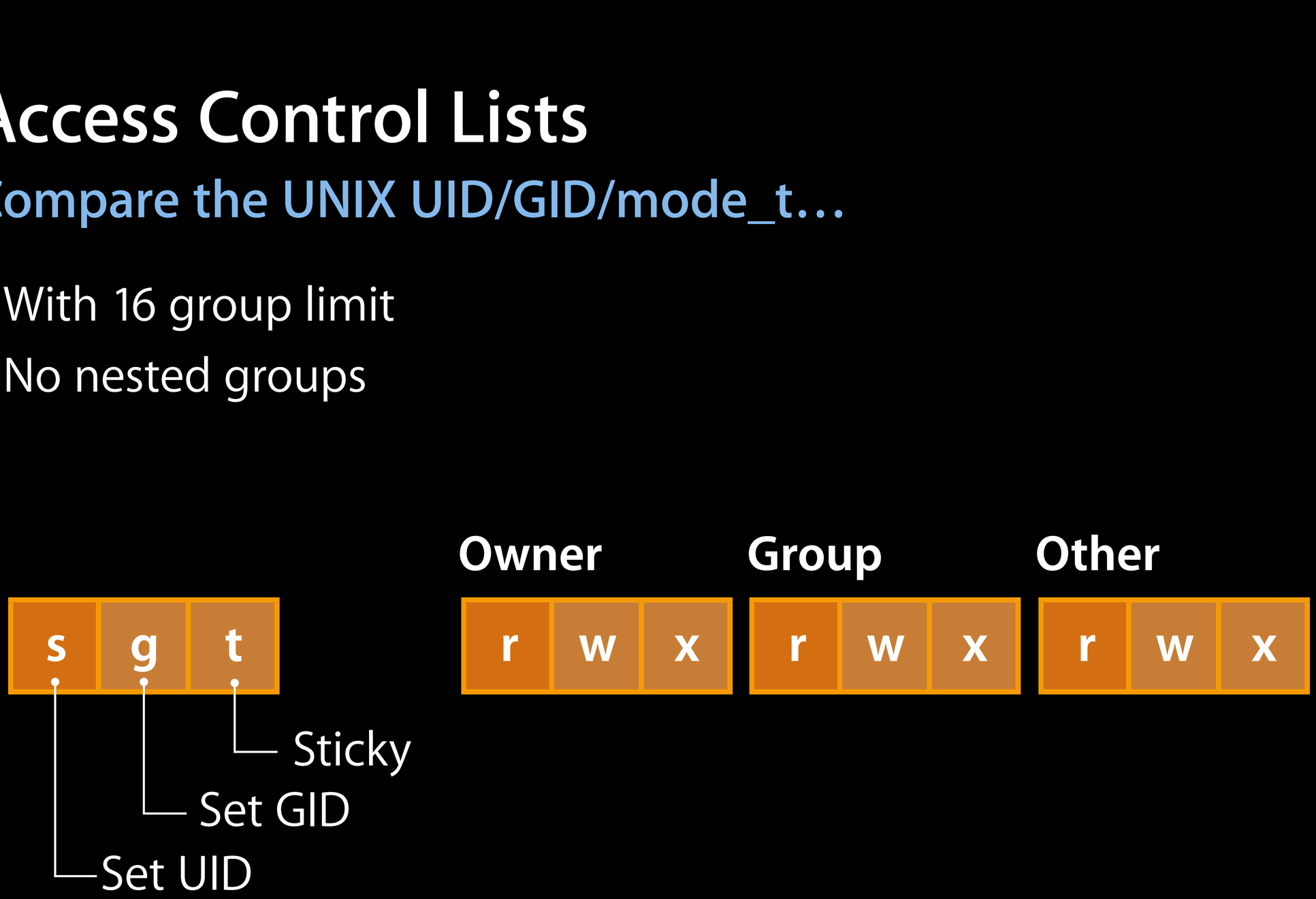
Terminology

ACL	Access Control List
ACE	Access Control Entry
UID	User ID
GID	Group ID
GUID/UUID	Globally/Universally Unique ID
SID	NT Security ID

Access Control Lists

Compare the UNIX UID/GID/mode_t…

- With 16 group limit
- No nested groups



Access Control Lists

…with the finer granularity of ACLs

File	Directory
read file	read directory
write file	add file entry to directory
execute file	search in directory
delete file	delete directory
append to file	add subdirectory entry to directory
N/A	remove file entry from directory
read extended (named) attributes	read extended (named) attributes
write extended (named) attributes	write extended (named) attributes
read basic attributes	read basic attributes
write basic attributes	write basic attributes
read file permissions/ACL	read directory permissions/ACL
write file permissions/ACL	write directory permissions/ACL
take ownership	take ownership

Access Control Lists

Overview

- ACLs made up of Access Control Entries (ACEs)

```
# ls -le
-rw-r--r--+ 1 emura  staff  0 Apr 28 14:06 foo
  1: guest deny read
  2: admin allow write,delete
  3: msmith inherited deny delete
  4: others inherited allow read
  5: admin inherited allow delete
  6: backup inherited deny read
  7: admin inherited allow write-security
```

- Static inheritance
- For any { **identity, operation, object** }, kauth will:
  - Scan/parse ACEs in order
  - Compare identity, resolve group(s) if needed
  - Mode_t and outstanding permissions

Access Control Lists

Overview

- Mac OS X server "only"
- fsaclctl
- Supported filesystems/network filesystem protocols:
  - HFS
  - SMB/CIFS
  - AFP
- Functionally close to Windows ACLs

Related session

Leveraging the Power of ACLs

Session 617
Wednesday 2:00pm

Access Control Lists

Command line

```
% /usr/sbin/fsaclctl -p /Volumes/depot -e
% rm -f foo ; touch foo
% ls -le foo
-rw-r--r--   1 emura  staff  0 Apr 27 12:40 foo

% chmod +a "admin allow read" foo
% ls -le foo
-rw-r--r-- + 1 emura  staff  0 Apr 27 12:40 foo
 0: user:admin allow read
```

Access Control Lists

API

```
/* check to see if ACLs are supported in the current directory*/
if (-1 == (retval = pathconf(".", _PC_EXTENDED_SECURITY_NP))) {
   err(1, "pathconf()");
} else {
   if(0 == retval) {
      fprintf(stderr,"ACLs not supported here (retval=%ld)\n",
      retval);
      exit(1);
   }
}
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
if (0 != mbr_uid_to_uuid(getuid(), *uuid)) {
        perror("mbr_uid_to_uuid()");
        free(uuid);
        exit(1);
}

result = acl_readonly_example(uuid);
free(uuid);
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
int
acl_readonly_example(uuid_t *uuid)
{
.
.
.

        /* initialize our ACL */
        if (NULL == (acl = acl_init(32)))
                err(1, "acl_init()");
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
/*
 * create an ACE
 *
 * acl_create_entry_np() has a position capability via the
 * 'entry_index' argument (ACL_FIRST_ENTRY or ACL_LAST_ENTRY)
 */
if (0 != acl_create_entry(&acl, &ace))
        err(1, "acl_create_entry()");
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
/* allow or deny */
if (0 != acl_set_tag_type(ace, ACL_EXTENDED_ALLOW))
        err(1, "acl_set_tag_type()");

/* associate this with our uuid */
if (0 != acl_set_qualifier(ace, uuid))
        err(1, "acl_set_qualifier()");
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
/* grant "read only" permissions */
if (0 != acl_get_permset(ace, &perms))
        err(1, "acl_get_permset()");

if (0 != acl_clear_perms(perms))
        err(1, "acl_clear_perms()");

if (0 != acl_add_perm(perms, ROPERMS))
        err(1, "acl_add_perm()");

if (0 != acl_set_permset(ace, perms))
        err(1, "acl_set_permset()");
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
#include <membership.h>
#include <sys/acl.h>


#define ROPERMS ( ACL_READ_DATA | ACL_READ_SECURITY |\
                  ACL_READ_ATTRIBUTES | ACL_READ_EXTATTRIBUTES )


int
acl_readonly_example(uuid_t *uuid)
{
```

From ‘acl_api_fragment.c’ sample code

Access Control Lists

API

```
/* create a file security object */
fsec = filesec_init();

/* add the ACL to the security descriptor */
filesec_set_property(fsec, FILESEC_ACL, &acl);
acl_free(acl);

/* turn off all other permissions on the file */
filesec_set_property(fsec, FILESEC_MODE, 0);

/* create a file using our ACL */
int result = openx_np("foo", O_CREAT|O_EXCL|O_RDWR, fsec);
```

From ‘acl_api_fragment.c’ sample code

File Notification

http://developer.apple.com/wwdc2005/

File Notification

kqueues

- More efficient than `select() & poll()`
- Great for directory update notification
- Current limitations
  - Not supported on all volume formats
  - Local only (events on server may go unnoticed)
  - No payload
  - Recursive directory hierarchy challenge

File Notification

kqueue(2)

```
if ((kq = kqueue()) == -1)
        errx(3, "cannot create kqueue");

path = strdup(argv[i]);

if (-1 == (fd = open(path, O_EVTONLY)))
        err(2, "open() on %s failed", path);

EV_SET(&kev, fd, EVFILT_VNODE, EV_ADD|EV_CLEAR,
        MY_NOTES, 0, NULL);
```

From ‘kqueue_fragment.c’ sample code

File Notification

kqueue(2)

```
if (kevent(kq, &kev, 1, NULL, 0, NULL) == -1)
        err(4, "cannot add kev to queue");

while (1) {
        fprintf(stderr, "calling kevent\n");
        nevents = kevent(kq, NULL, 0, &kev, 1, NULL);

        if (nevents == -1)
        err(5, "cannot wait for event");

        if (nevents > 0) {
                printf("ident = %ld  ", kev.ident);
```

From ‘kqueue_fragment.c’ sample code

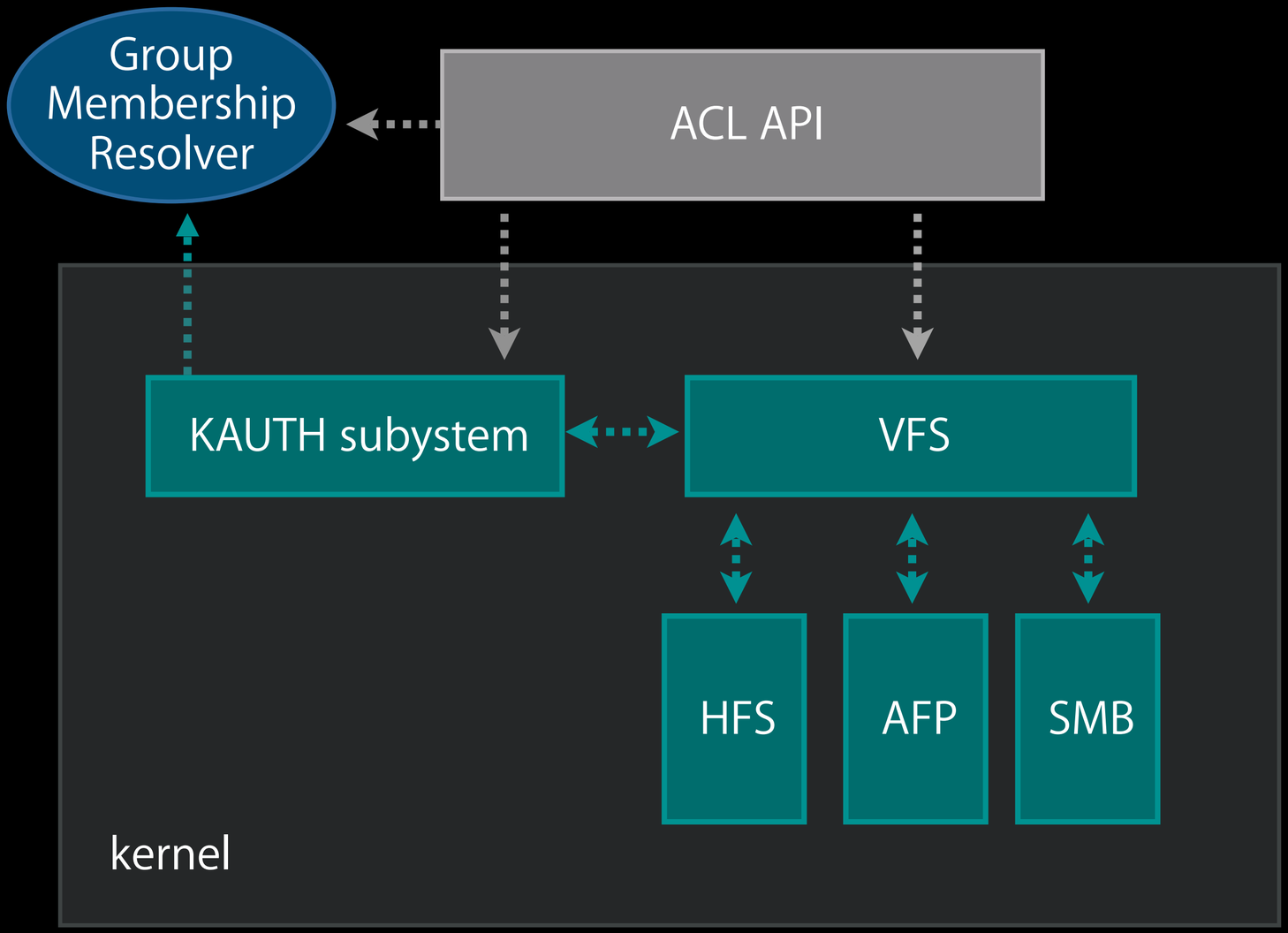
File Notification

kauth

- A kernel framework for authorization decision making
- Not initially designed for file/directory notification

File Notification

The kauth subsystem



File Notification

kauth

- Plugins for 3rd party developers exist and serve two purposes:
  - Provide mechanism for 3rd party code to participate in the authorization of filesystem operations
  - Provide notification of the performance of possibly-authorized operations

File Notification

kauth plugins

- Scopes and actions
- Simple voting mechanism
  - Anyone may veto
  - One must agree
  - Anyone may abstain
- Your plugin typically permitted to block, but be aware of
  - Hot paths
  - Recursions / deadlocks

Copy Engine

http://developer.apple.com/wwdc2005/

Copy Engine

FSCopyObjectAsync()

```
 if (copy)
   err = FSCopyObjectAsync(fileOp, source, destDir, destName, flags,
callback, statusChangeInterval, contextString != NULL ?
&clientContext : NULL);
 else
   err = FSMoveObjectAsync(fileOp, source, destDir, destName, flags,
callback, statusChangeInterval, contextString != NULL ?
&clientContext : NULL);

 return err;
```

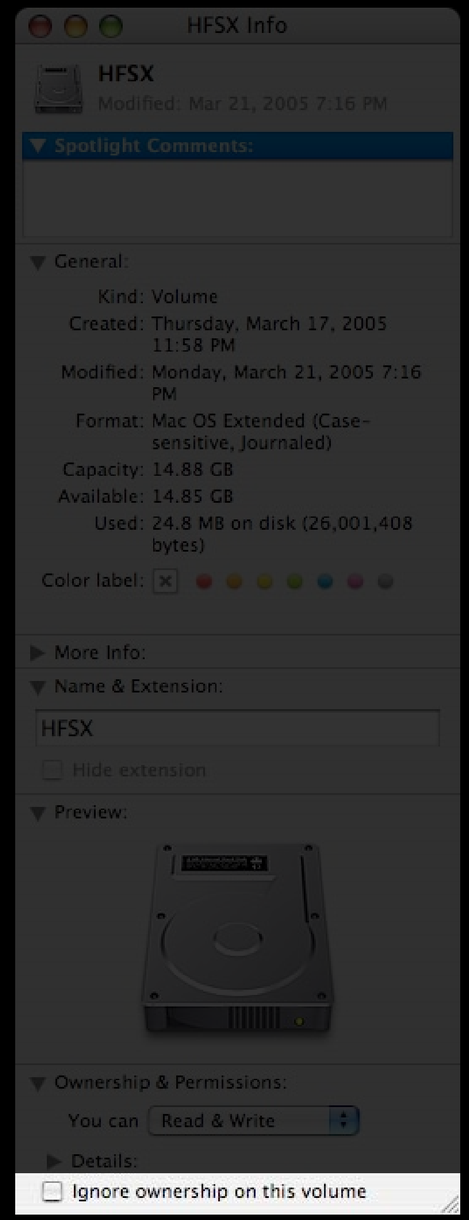
From ‘FSFileOperation’ sample code

Oldies but Goodies

http://developer.apple.com/wwdc2005/

UID 99

- The “Ignore ownership on this volume” check box
- Set by default for drives not present at first boot after an install
- Corresponds to MNT_IGNORE_OWNERSHIP (mount_t)
- mount_hfs -u and -g options can add a wrinkle



So let’s walk through the possible configurations…

UID 99

In the case where MNT_IGNORE_OWNERSHIP is not set and no explicit owner is provided at mount time

- UID/GID reported will be the actual UID/GID on disk
- UID/GID written will be the traditional user and parent directory group
- Items with UID 99 will be considered owned by the current user for authorization purposes

UID 99

For volumes mounted MNT_IGNORE_OWNERSHIP and without an explicit owner provided at mount time

- UID/GID reported will be 99/99 regardless of what is on disk
- UID/GID written will be 99/99
- Items will be considered owned by the current user for authorization purposes

UID 99

For volumes mounted MNT_IGNORE_OWNERSHIP and with an explicit owner provided at mount time

- UID/GID reported will be the mount-time owner/group
- UID/GID written will be 99/99
- Items will be considered owned by the provided owner/group for authorization purposes

UID 99

In the case where MNT_IGNORE_OWNERSHIP is not set and an explicit owner is provided at mount time

- UID/GID reported will be the actual UID/GID on disk
- UID/GID written will be the traditional user and parent directory group
- Items with UID 99 will be considered owned by the provided user for authorization purposes

Miscellaneous…

- fsync() and fcntl(F_FULLFSYNC)
  - Misconceptions
  - Performance implications
  - See fsync(2) and fcntl(2) man page for more detail

- HFSX (“Default.png” vs. “default.png” incidents)

Tools

- `fs_usage`
  - Filters for specific process or filesystem
  - DYLD_IMAGE_SUFFIX and Carbon FileManager
- `sample`
- `lsof`
- `ktrace`
- General profiling tools like “Shark”

Demo

http://developer.apple.com/wwdc2005/

More Information

http://developer.apple.com/wwdc2005/

Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

Related Sessions

Leveraging the Power of ACLs	Session 617 Wednesday 2:00PM
Writing Filesystems Using Supported KPIs	Lab 5001 Wednesday 3:30PM

Who to Contact

Deric Horn
Topic Manager
deric@apple.com

Chris Emura
Filesystem Engineering Manager
cemura@apple.com

Q&A

http://developer.apple.com/wwdc2005/

WWDC2005

Worldwide Developers Conference